

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**SciVerse ScienceDirect**

Physics Procedia 33 (2012) 975 – 982

Physics

**Procedia**

2012 International Conference on Medical Physics and Biomedical Engineering

# Double Motor Coordinated Control Based on Hybrid Genetic Algorithm and CMAC

Shaozhong Caoa, Ji Tub

*aSchool of Information & Mechanical Engineering, Beijing Institute of Graphic Communication, Beijing, China  
cszh6502@163.com**bInstitute of Automation, Chinese Academy of Sciences, Haidian District, Beijing, China  
tj100@163.com*

---

## Abstract

A novel hybrid cerebellar model articulation controller (CMAC) and online adaptive genetic algorithm (GA) controller is introduced to control two Brushless DC motor (BLDCM) which applied in a biped robot. Genetic Algorithm simulates the random learning among the individuals of a group, and CMAC simulates the self-learning of an individual. To validate the ability and superiority of the novel algorithm, experiments have been done in MATLAB/SIMULINK. Analysis among GA, hybrid GA-CMAC and CMAC feed-forward control is also given. The results prove that the torque ripple of the coordinated control system is eliminated by using the hybrid GA-CMAC algorithm.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of [name organizer]

Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/3.0/).

Keywords : CMAC, Genetic Algorithm, BLDCM, Coordinated Control

---

## 1. Introduction

With its excellent static and dynamic speed regulating performance, Brushless DC Motor (BLDCM) is widely used in biped robot <sup>[1]</sup>. It's also architecturally simple, reliable, and easy to control. When the robot multi-axis control system needs to coordinate, the second motor must synchronize with the first motor. As parameters changing when motors are running, the torques of the loads is varying. The speed coordination between the two motors will be subject to interference <sup>[2]</sup>. However, the genetic algorithm <sup>[3]</sup>

---

Supported by Beijing Natural Science Foundation.(4092013)

(GA), particle swarm optimization<sup>[4]</sup>, neural network<sup>[5]</sup> and fuzzy control<sup>[6]</sup> can be applied to help the traditional PID controller. With the trigger voltage controlled by these new algorithms, the speed of the second motor follows the first motor quickly and precisely.

Within the above algorithms, genetic algorithm is an advanced heuristic search optimization method with the advantages of simple, robust and doesn't need any initial information to find the global optimization. But as a stochastic simulation of biological population learning algorithm which is weak in local search and slow in convergence, genetic algorithm must be improved in stability and accuracy<sup>[7]</sup>. In contrast, the cerebellar model articulation controller (CMAC) can simulate the individual of a population. The advantages of CMAC are that it has strong local generalization ability and it is fast in convergence. Its minimum value is unique. Due to less iteration in comparison with other neural networks, its program executes faster so that CMAC can lend itself to real-time control<sup>[8]</sup>. But the CMAC needs some initial information to control a system in the beginning. Combine these two algorithms and learn from each other, we can improve the performance of coordinated control.

## 2. Mathematical model of BLDCM

Because of the gap magnetic field, back electromotive force (EMF) and stator current of a Brushless DC motor are non-sinusoidal; the modeling of BLDCM is difficult. To simplify the analysis, some suppositions are made:

- Three-phase stator windings are completely symmetry and the corresponding time constant is negligible.
- Ignore the stator current flux, slot effect and saturation on the impact of air-gap flux.
- Do not consider the temperature effect or other factors impact on the motor parameters.

The dynamic characteristic equations of Brushless DC motor can be described as<sup>[9]</sup>:

$$U(t) = E_a(t) + R \cdot i(t) + L \cdot di(t) / dt \quad (1)$$

$$E_a(t) = K_e \cdot n(t) \quad (2)$$

$$T_a(t) = K_t \cdot i(t) \quad (3)$$

$$T_a(t) = J \cdot dn(t) / dt + D \cdot n(t) \quad (4)$$

where  $U$  is the trigger voltage,  $E_a$  is the back-EMF,  $R$  is the three-phase stator resistance,  $L$  is the armature inductance,  $I$  is the phase current,  $K_e$  is the EMF constant,  $K_t$  is the torque coefficient,  $n$  is the motor speed,  $T_a$  is the motor torque,  $J$  is the inertia coefficient,  $D$  is the friction coefficient.

Carrying out Laplace transform with (1) ~ (4), then there are the following equations below.

$$U(s) = E_a(s) + R \cdot I(s) + L \cdot s \cdot I(s) \quad (5)$$

$$E_a(s) = K_e \cdot n(s) \quad (6)$$

$$T_a(s) = K_t \cdot I(s) \quad (7)$$

$$T_a(s) = J \cdot s \cdot n(s) + D \cdot n(s) \quad (8)$$

In two-axis robot coordinated control system, the parameters of the second motor is:  $R=5\Omega$ ,  $K_e=0.125\text{V}\cdot\text{s}/\text{rad}$ ,  $K_t=15\text{Nm}/\text{A}$ ,  $L=5\times 10^{-3}\text{H}$ ,  $D=0.01\text{Nm}\cdot\text{s}/\text{rad}$ ,  $J=0.03\text{kg}\cdot\text{m}^2$ . The transfer function between the speed and the PWM voltage of the second motor is obtained from (5) ~ (8). As follow,

$$\frac{n(s)}{U(s)} = \frac{15}{0.00015s^2 + 0.15s + 1.925} \quad (9)$$

Using discrete control in the computer control system, the z-transform transfer function which yields form (9) is

$$\frac{n(z)}{U(z)} = \frac{0.03675z + 0.02639}{z^2 - 1.36z + 0.3679} \quad (10)$$

### 3. Genetic algorithm parameter optimization

Genetic algorithm is an evolution computing algorithm which simulates Darwin's natural selection and genetic mechanism. It is a means for search global optima by simulate the natural evolution process. It was originally introduced by Professor J.Holland in 1975 at the University of Michigan. His book named 'Adaptation in Natural and Artificial Systems' is quite famous. The genetic algorithm which Professor J.Holland proposes is typically called the simple genetic algorithm.

This article addresses the adaptive online GA to self-turning PD parameter. At sample time k, selects enough individual, calculates the fitness of different individuals. By optimizing, GA selects the PD parameters corresponding with the most fits individual. In order to obtain a smoothly coordinated control process and prevent from overshoot, we accumulate the absolute error and the error changing rate to consist a minimum objective function<sup>[10]</sup>. At sample time k, the minimum objective function of individual i is

$$J(i) = \theta_1 * |error(i)| + \theta_2 * |\Delta e(i)| \quad (11)$$

Where, error(i) is the speed tracking error of individual i at sample time k,  $\Delta e(i)$  is the changing rate of the speed tracking error;  $\theta_1, \theta_2$  is the experience coefficient, respectively.

Use a penalty function to avoid overshoot. That is to say, when error (i) < 0, put the overshoot as a term of the minimum objective function

$$J(i) = J(i) + 100 * |error(i)| \quad (12)$$

In this paper, we use the adaptive on-line GA instead of the simple GA. The adaptive GA changes the mutation probability  $P_m$  automatically. The bigger the fitness, the smaller the mutation probability. As calculate in the formula

$$P_m = 0.2 - [1 : pop] * 0.01 / pop \quad (13)$$

Where, pop is the number of the population.

The steps of using GA to optimize PD parameters are as follows:

*Step1:* Determine the approximate range of PD parameter, and then coding.

*Step2:* Initialize the fitness value; randomly generate a population which consists of n individuals.

*Step3:* Decode the parameters correspond with the individual. Use this parameter to calculate the cost function J and fitness function f, where  $f = 1/J$ ;

*Step4:* Reproduce, crossover and mutation the population, produce the next generation.

*Step5:* If the parameters convergent or reach the pre-set target, output the optimal value. Otherwise, return to step (3) and (4).

#### 4. Hybrid ga-CMAC controller

##### 4.1. CMAC feed-forward control system

CMAC is an adaptive neural network which can express complicated non-linear function by looking up the table. It can change the values in the table as well as sort the information in it. CMAC has better non-linearization approach ability than normal neural networks, which is suited for complex non-linear dynamic real-time control<sup>[11]</sup>.

The schematic of the traditional CMAC feed-forward control system is described in Fig.1.

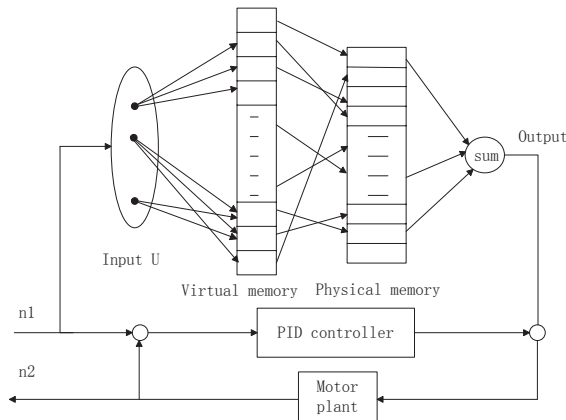


Fig. 1 The traditional CMAC feed-forward control system

Fig.1 denotes a parallel CMAC and PID control system, which CMAC is a feed-forward controller. The concept of CMAC feed-forward controller is that: mapping the input in virtual memory, and found corresponding state address in physical storage unit. The output of the CMAC is the sum of the network weights which stored in the physical memory. Compare the CMAC output with the output of the hybrid controller, and then modify those weights according to the gradient descent direction.

##### 4.2. Genetic algorithm instead of PID

To improve the robot coordinated control system, using the genetic algorithm controller to replace traditional PID controller. The schematic of double-motor coordinated control system is given in Fig.2.

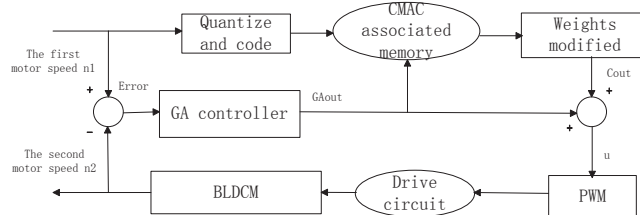


Fig. 2 Schematic of double motor coordinated control system

The one-dimensional output of CMAC controller is

$$Cout(k) = \sum_{i=1}^{L+f} w_i \delta_i \quad (14)$$

Where,  $\delta_i$  is the binary operator,  $w_i$  is the network weight,  $f$  is the generalization parameter.

The input of CMAC  $U_j$  is the speed of the first motor  $n_1$ . Partition the input  $U_j$  into  $(L+2f)$  quantitative interval within the interval  $[U_{\min}, U_{\max}]$ . The following equations are introduced:

$$u_1, u_2, \dots, u_f = U_{\min} \quad (15)$$

$$u_j = u_{j-1} + \Delta u \quad (j=f+1, \dots, f+L) \quad (16)$$

$$u_{L+f+1}, \dots, u_{L+2f} = U_{\max} \quad (17)$$

Where,  $\Delta u$  is the generalization constant which is bigger than quantitative interval and usually assume as

$$\Delta u = (U_{\max} - U_{\min}) / (L-1) \quad (18)$$

The weights of CMAC are available only in the generalization interval, so that the binary operator  $\delta_i$  is

$$\delta_i = \begin{cases} 1 & U_j \in [u_j, u_{j+f}], j = 1, \dots, f+L \\ 0 & \text{others} \end{cases} \quad (19)$$

The output of the hybrid controller  $u(k)$  is the sum of the GA controller and the CMAC controller.

$$u(k) = GAout(k) + Cout(k) \quad (20)$$

Where,  $GAout(k)$  is the output of the GA controller.

In each control cycle, compare the output of the CMAC and the output of the hybrid controller  $u(k)$  to modify the weights. The weights adjustment index of CMAC is

$$E(k) = (Cout(k) - u(k))^2 / (2f) \quad (21)$$

Substitute (14) and (20) into (21), we obtain the modified value of the CMAC weights.

$$\Delta \omega(k) = -\eta \cdot dE(k) / d\omega = \eta \cdot \delta_i \cdot GAout(k) / f \quad (22)$$

Where,  $\eta \in (0, 1)$  is the learning rate of the weights.

In order to avoid the oscillation in training process and accelerate the convergence rate, we consider the gradient direction both in time  $(k-1)$  and time  $(k)$ , and add the momentum term. The rule for weights modifying is

$$\omega(k) = \omega(k-1) + \Delta \omega(k) + \alpha \cdot (\omega(k) - \omega(k-1)) \quad (23)$$

Where,  $\alpha \in (0, 1)$  is the momentum coefficient.

When the system runs, initialize  $\omega$  with zero. The output of CMAC is zero, too. Genetic algorithm controller searches the global optimal PD parameters. As the CMAC learning, the output of genetic algorithm PD controller gradually close to zero, then the weights of CMAC becomes stable. The CMAC output approaches to the hybrid output  $u(k)$ . The output  $u(k)$  is the PWM trigger voltage of the second motor. It precisely changes the speed of the second motor through the driving circuit to coordinate with the first motor.

## 5.Simulation and Analysis

### 5.1.Simulation

In the purpose of analysis the effect of the hybrid GA-CMAC controller, simulations have been done in the MATLAB/SIMULINK version 2009a environment. The GA controller and the CMAC feed-forward controller are also simulated in MATLAB. The three different controllers are simulated under the same settings to compare the performance of speed coordinated control. The Hybrid GA-CMAC, GA, and CMAC feed-forward control simulation model are presented in Fig.3.

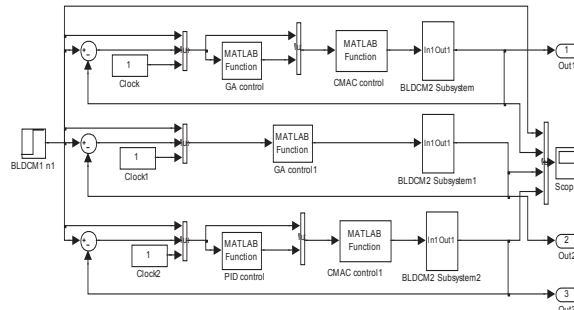


Fig. 3 Simulation model of GA, GA-CMAC and CMAC

Use s-functions to implement the design of discrete controllers. Initialize the error through the clock function, so as to achieve the integral and differential of the errors. Simulation time is set to 1s, sampling time is 0.001s. The speed of the first motor is a step signal which magnitude is 1750 r/min.

The real-number encoding was used in GA program. The parameter settings of GA program are listed in table 1.

TABLE I PARAMETERS SETTINGS IN GA S-FUNCTION

$\theta_1$	0.95	$\theta_2$	0.05
pop	120	Pc	0.9

In order to avoid the parameters selection area becomes too large, set  $K_p$  and  $K_d$  under experience first. Then the GA is used to reduce the random test on looking for the optima parameters.

The parameters settings of CMAC program are given in table 2.

TABLE II PARAMETERS SETTINGS IN CMAC S-FUNCTION

L	100	f	5
$\alpha$	0.04	$\eta$	0.1

In CMAC feed-forward s-function,  $K_p$  is 23 and  $K_d$  is 5.

### 5.2.Analysis among the three controllers

In the simulation, the results of GA, Hybrid GA-CMAC and CMAC feed-forward control are displayed in Fig.4.

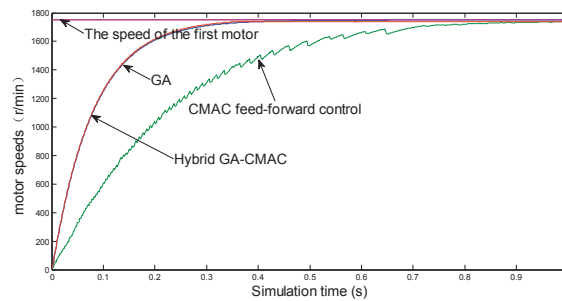


Fig. 4 The coordinated control results of the three controllers

Fig.4 shows that the torque ripple of the second motor is large when using the CMAC feed-forward controller, and the settling time is also too long, so that it can not coordinate with the first motor smoothly and quickly.

For further analysis between GA and Hybrid GA -CMAC, we change the Y-axes in Fig.4 through the axes properties in SIMULINK figure edit option. The more helpful figure for analysis is exhibited in Fig.5.

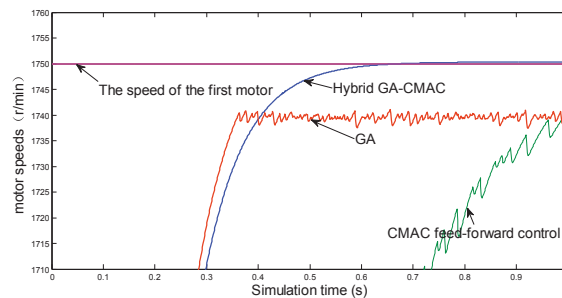


Fig. 5 The coordinated control results of the three controllers ( a part of Fig.4 )

In Fig.5, it presents that both the GA and the Hybrid GA- CMAC can quickly react when the input signal changes. But compared with the Hybrid controller, the system use the GA controller has large torque ripple after 0.33 second and it can not track the speed of the first motor exactly. Its maximum error is about 10r/min, while the error of the hybrid controller is less than 2r/min. For the sake of the GA controller can not quickly convergent to a minimum value, it is not helpful in the real coordinated control system. However, thanks to its high generalization ability, the motor use the Hybrid GA-CMAC controller can not only precisely coordinate with the first motor, but also has no torque ripple. It is of great significance in practical applications.

Learning form the simulation experiments, the proportional coefficient is decisive in GA controller, while in the Hybrid GA-CMAC controller,  $K_p$  affects the controller little. That is to say, the robustness of the system has been improved.

The simulation experiments also demonstrate that, the work of finding PID parameters is tremendous blind and cumbersome. Under the help of genetic algorithm, human can be in some sense liberated from the bother work of tuning PID parameters.

## 6.Conclusions

Attributing the global search ability to GA and the local generalization ability to CMAC, the torque

ripple has been eliminated when using the Hybrid GA-CMAC controller in double motor coordinated control system. Furthermore, the new controller needs little work to tuning the PID parameters, which is more intelligent. Analyses of the three controllers demonstrate that use the CMAC can solve the weakness of slow convergence. In the mean time, GA is effective in global variable optimization.

## References

- [1] Cao Shaoyong, Yang Guowei, Tu Xuyan, "Current Status of SoftMan Research and Its Application in Publishing and Printing Industry," *Networking, Sensing and Control*, 2008. ICNSC 2008, IEEE International Conference on 6-8, pp.1243 - 1246, April 2008
- [2] Xiao-Dong Zhang, "Simulation Analysis on Differential Speed Drive of Double BLDCM Based on PID," *Innovative Computing Information and Control*, 2008. ICICIC '08, pp. 285 - 285, June 2008.
- [3] Alen VarSek, Tanja Urbantit, and Bodgan Filipit, "Genetic Algorithms in Controller Design and Tuning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no.5, pp.1331-1339, September/October 1993
- [4] Yu Ren, Xiaobai Xu, "Optimization Research of PSO-PID Algorithm for the Design of Brushless Permanent Magnet Machines," *Fifth IEEE International Symposium on Embedded Computing*, pp. 26-29, 2008
- [5] Kazushige Koiwai, Kazuo Kawada and Toru Yamamoto, "Design of a Human-Skill Based PID Controller using CMACs," *International Conference on Control, Automation and Systems* 2008, pp.64-68, Oct, 2008
- [6] D.Devaraj, B.Selvabala, "Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional–integral–derivative controller in automatic voltage regulator system," *IET Gener. Transm. Distrib.*, vol. 3, Iss. 7, pp. 641–649, 2009
- [7] Shiu Yin Yuen, Chi Kin Chow, "A Genetic Algorithm That Adaptively Mutates and Never Revisits.," *IEEE transactions on evolutionary computation*, vol. 13, no. 2, pp.454-472, 2009
- [8] Miao Gao, Yanpeng Wang, "Brushless Direct Current Motor Control System Based on Digital Signal Processor," *Information Engineering*, 2009. ICIE '09. vol. 1, pp.181 - 184 ,July 2009
- [9] Gwo-Ruey Yu and Rey-Chue Hwang, "Optimal PID Speed Control of Brushless DC Motors Using LQR Approach," 2004 *IEEE International Conference on Systems, Man and Cybernetics* ,TaiWan, pp.473-478
- [10] Cao Shaoyong, Yang Guowei, Tu Xuyan, "Approximate Solution Research of State Equation for Controlled Nonlinear Systems," *Networking, Sensing and Control*, 2008. ICNSC 2008, pp.1784 – 1787, April 2008
- [11] W.Thomas Miller, Felson H.Glanz, L. Gordon Kraft, "CMAC:An Associative Neural Network Alternative to Backpropagation," *Proceedings of the IEEE*,vol,78,no.10, pp.1561-1567, 1990